

ANALYZING EXPLICITLY-STRUCTURED DISCOURSE IN A LIMITED DOMAIN: TROUBLE AND FAILURE REPORTS*

Catherine N. Ball
Unisys Corporation
Paoli Research Center
P.O. Box 517, Paoli PA 19301

ABSTRACT

Recent theories of focusing and reference rely crucially on discourse structure to constrain the availability of discourse entities for reference, but deriving the structure of an arbitrary discourse has proved to be a significant problem. A useful level of problem reduction may be achieved by analyzing discourse in which the structure is explicit, rather than implicit. In this paper we consider a genre of explicitly-structured discourse: the Trouble and Failure Report (TFR), whose structure is both explicit and constant across discourses. We present the results of an analysis of a corpus of 331 TFRs, with particular attention to discourse segmentation and focusing. We then describe how the Trouble and Failure Report was automated in a prototype data collection and information retrieval application, using the PUNDIT natural-language processing system.

INTRODUCTION

Recent theories of focusing and reference rely crucially on discourse structure to constrain the availability of discourse entities for reference, but deriving the structure of an arbitrary discourse has proved to be a significant problem ([Webber 88]). While progress has been made in identifying the means by which speakers and writers mark structure ([Grosz 86], [Hirschberg 87], [Schiffrin 87], [Webber 88]), much work remains to be done in this area.

As is well known, initial progress in computational approaches to syntax and semantics was facilitated by reducing the problem space to discourses in technical sublanguages, in simplified registers, in restricted domains¹. For Computational Pragmatics, the analysis of explicitly-structured discourse can provide a similar level of problem reduction. By removing the theoretical obstacle of deriving discourse structure, we can more readily evaluate the effect of this structure on focusing and reference.

In this paper we consider a genre of explicitly-structured discourse, namely the 'form', in which each labelled box and the response within it constitute a discourse segment. From the perspective of discourse understanding, the study of forms-discourse offers considerable advantages: the structure of the form is pre-defined and constant across discourses, and it is possible to study patterns of reference in narrative responses without excessive reliance on intuition. The particular form which we consider here is the Trouble and Failure Report (TFR). We first discuss the results of an analysis of 331 TFRs, and then describe the implementation of a TFR analysis module using the PUNDIT natural-language processing system.

THE TROUBLE AND FAILURE REPORT

TFRs are used to report problems with hardware, software, or documentation on equipment on board Trident and Poseidon submarines. These reports originate on board the submarine, and those concerning the Navigational Subsystem (which is managed by the Unisys Logistics Group) are routed to Unisys for analysis and response.

*This work has been supported by DARPA contract N00014-85-C-0012, administered by the Office of Naval Research.

¹See for example the papers in [Grishman 86b].

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 1989		2. REPORT TYPE		3. DATES COVERED 00-00-1989 to 00-00-1989	
4. TITLE AND SUBTITLE Analyzing Explicitly-Structured Discourse in a Limited Domain: Trouble and Failure Reports				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Unisys Corporation,Center for Advanced Information Technology,PO Box 517,Paoli,PA,19301				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 6	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

The TFR contains a formatted section and up to 99 lines of free text. The formatted section includes coded information identifying the message originator, date, equipment, and failed part. The free text is divided into 5 sections, labelled A-E, each of which documents a specific aspect of the problem being reported. A sample hardware TFR is given below²:

<Formatted lines...>

A. WHILE PERFORMING SDC 955Z (GENERATION OF LASER BEAMS) TRANSPORTER UPPER TRANSLOCK WENT OFF LINE. B. UPPER TRANSLOCK INTERPORT SWITCH WENT BAD, UNABLE TO RE-ENERGIZE ETHER REGULATOR IN UPPER TRANSLOCK WHEN INTERPORT SWITCH DEPRESSED. C. DETERIORATION DUE TO AGE AND WEAR. D. REPLACED INTERPORT SWITCH WITH A NEW ONE FROM SUPPLY. E. NONE.

TFRs are stored in a historical database. Although the formatted data can be mapped to specific fields of database records, which can then be accessed by a query language, the free-text portions are stored as undigested blocks of text. Currently, keyword search is the only method by which the text can be accessed. Problems with keyword search as a method of information retrieval are well-known³, and this is an area in which NLP techniques can be applied, with potential benefits of increasing the efficiency and accuracy of information retrieval.

As part of an internally and DARPA-funded R&D project, we applied PUNDIT ([Grishman 86a], [Dahl 87], [Dahl 86]) to the analysis of TFRs. Previous applications of PUNDIT to the analysis of the **Remarks** field of Navy messages had required only a superficial level of discourse processing above the paragraph. But the richer discourse structure of TFRs required a more sophisticated approach, including a discourse interpretation module and a segment-based approach to focusing. But although the discourse structure of TFRs forced a number of issues, the fact that this structure is explicit and constant across discourses greatly facilitated the analysis of TFR discourse, to which we now turn.

TFRS AS DISCOURSE

The perspective of a sentence-based grammar might lead us to ignore the formatted lines of a TFR, to consider as discourse only the textual portions, and to interpret each element of the latter as a full or a 'fragmentary' sentence (cf. [Linebarger 88]). On this approach, we would be prepared to analyze the following TFR extract as discourse:

WHEN ATTEMPTING TO ERASE 2 METERS ON THE EVENT RECORDING STRIP, THE STRIP WOULD CONTINUOUSLY RUN. INVESTIGATION REVEALED THAT "NOYB" WAS BEING GENERATED. AGE AND USE. REPLACED WITH NEW ITEM AND RETURNED OLD TO SUPPLY. NONE.

However, it is immediately apparent that this approach would be incorrect: the discourse is incoherent.

Two distinct problems may be identified. After the first two sentences, the remainder bear no apparent relation to preceding discourse. Secondly, one or more discourse entities appear to be missing: age and use – of what? Who (or what) replaced what with a new item? None – of what?

The source of incoherency is two-fold: we are missing the initial context established by the interpretation of the formatted lines of the TFR, and we have ignored the basic unit of TFR discourse: the REQUEST-RESPONSE PAIR. As it turns out, each of the elements of the formatted lines (henceforth the *header*) has a positional interpretation, and each of the labels A-E maps to a noun phrase label. Each label can be interpreted as a request for information. Now reconsider the TFR above in this light:

TFR number	:1234567
Equipment code	:TRANSPORTER
Part number	:01223426

²As we are not permitted to cite data from actual TFRs, all examples in this paper are purely fictional. However, the crucial linguistic properties have been preserved.

³But a recent study has shown them to be even more serious than users of keyword systems might have realized ([Blair 85]).

Date of Trouble :1/21/89
Report date :2/15/89
Originator :JONES

- A. First indication of trouble: WHEN ATTEMPTING TO ERASE 2 METERS ON THE EVENT RECORDING STRIP, THE STRIP WOULD CONTINUOUSLY RUN.
- B. Part failure: INVESTIGATION REVEALED THAT "NOYB" WAS BEING GENERATED.
- C. Probable cause: AGE AND USE.
- D. Action taken: REPLACED WITH NEW ITEM AND RETURNED OLD TO SUPPLY.
- E. Remarks: NONE.

The discourse is now coherent. As can be seen, responses are interpreted relative to their labels, not to each other. The previously missing discourse entity for the referent of **NONE** is evoked by the label **Remarks** (i.e., *No remarks*), what was replaced is the failed part (identified by the part number), it is the speaker (JONES) who replaced it, and finally, the implicit argument of **AGE AND USE** is that same failed part.

These results underline the need to consider the entire TFR as discourse, and to provide an account of the request-response pair as the basic unit of TFR discourse. In the following sections, we sketch such an account, and then turn to the evidence for higher-level structure.

The Request-Response Pair

Between the request and the response a special type of cohesive relation ([Schiffrin 87]) exists, similar to that which binds question-answer pairs. In fact, we claim that at the level of discourse interpretation, the request and response form a discontinuous predicate-argument structure⁴. This view of the request-response pair arises from the need to account for the interpretation of pairs such as **Probable cause: BROKEN WIRE**, from which we are somehow able to conclude: *The respondent believes that a broken wire caused the failure.*

Very briefly, we suggest that the mechanisms required to achieve this result are essentially those required (at the level of sentence grammar) for the interpretation of specificational copular sentences⁵: lambda-abstraction, function application, and lambda-reduction. First, we take the heads of NP labels to be relational nouns with internal argument structure. For both (1a) and (1b) below, we derive the representation in (2) by lambda-abstraction on the free variable. Function application and lambda-reduction yield the representation in (3), which is (non-coincidentally) also the representation of *A broken wire caused the failure*:

- 1a. The cause of failure was a broken wire.
- 1b. Cause of failure: broken wire
- 2. $[\lambda x[\text{cause}(x, \text{failure})]](\text{wire})$
- 3. $\text{cause}(\text{wire}, \text{failure})$

Discourse Segmentation, Focusing, and Reference

Each label in the TFR marks the start of a request-response pair. But does this unit correspond to a discourse segment, and if so, what is the higher-level structure of the TFR? We studied patterns of reference in TFRs and found evidence for both explicit and implicit structure, as described below.

The Role of the Message Header. The message header identifies the author of the report, the date on which it was sent, the date on which the problem occurred, the equipment, and the failed part. The dates are crucial to the temporal analysis of the message (which we shall not discuss here). Our analysis of the TFR corpus reveals the remaining entities (speaker, equipment, failed part) to be highly salient in the

⁴Specifically, we take the NP label to express an OPEN PROPOSITION ([Prince 86]), which can be viewed as an informationally incomplete predication; the response provides its argument.

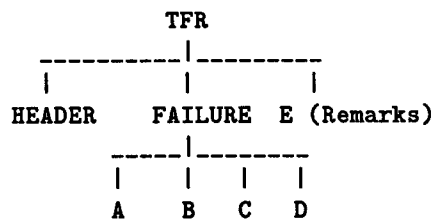
⁵See for example [Higgins 79] and [Delahunty 82].

discourse: they are available for pronominal reference in segments A-E, without requiring reintroduction by a full NP.

In addition, these entities fill implicit argument positions in the agentless passive, in possible intransitive uses of certain verbs (*replace*, *return*), and in some relational nouns (e.g. *age*, *wear*). These facts lead us to assign these three entities the distinguished status of *global foci*: entities which are always salient in the discourse context at the beginning of each new discourse segment.

Sections A-E. To determine whether each of these sections (**First indication of trouble**, **Part failure**, **Probable cause**, **Action taken**, **Remarks**) constitutes a discourse segment, we studied patterns of pronominal reference in the responses. The results were striking. In 804 occurrences of referential pronouns (707 of which were zero-subjects⁶), we found that only zero-subjects, *I*, *we*, and *this* refer beyond the boundary of the current request-response pair. 95% of the zero-subjects and all of the occurrences of *I* refer to the speaker. The remaining 5% of zero-subjects are distributed between reference to one of the global foci and segment-internal reference, with a slight bias towards the latter. *It*, *he*, *they*, *these*, *those* were found to refer purely locally (*that* did not occur). With the exception of *this* and the indexicals, pronominal reference is sensitive to the boundary of the request-response pair, and we conclude that each such pair is indeed a discourse segment.

In the demonstrative *this*, however, we found unexpected evidence for additional *implicit* structure: when occurring in segment E (**Remarks**), *this* can refer to the failure, or problem, described in segments A-D. Now, [Webber 88] argues that demonstrative reference of this type is sensitive to the *right frontier* of the discourse tree: that is, 'the set of nodes comprising the most recent closed segment and all currently open segments' (Webber 1988:114). If, as we had assumed, segments A-E are sisters, then segment D (**Action taken**) is the most recently closed segment, and there are no segments open other than the current segment, E. But none of the occurrences of *this* in segment E refer to segment D. To make sense of the data, we were led to the conclusion that segments A-D form an unlabelled, implicit segment: *the failure*. The **Remarks** segment is then the sister of this implicit segment; after closing segment D, this higher segment is closed, and thus lies on the right frontier when E is opened. From these observations we posit the following structure for the TFR:



THE TFR APPLICATION

The TFR application uses the PUNDIT natural-language processing system to analyze TFRs. The results of analysis are passed to a database module, which maps PUNDIT's representations to pre-defined records in a Prolog relational database. This database can then be queried using a natural-language query facility (QFE). Here, we discuss only the analysis part of the application.

In terms of user interaction, the TFR data-collection program superficially resembles traditional data-processing approaches to forms automation: the system prompts for each item on the form, and the user's response to each prompt is validated. If the response is judged invalid, an error message is issued and the user is reprompted.

⁶ As in *INSTALLED NEW ITEM, RETURNED OLD TO SUPPLY*.

Under the covers, however, the approach is quite different: the data-collection program is in fact a discourse manager, controlling and interpreting a dialogue between itself and the user. As the dialogue proceeds, it maintains a model of the discourse, calls PUNDIT's syntactic and semantic/pragmatic components to analyze the user's responses, and then interprets the response in the context of the prompt to derive new propositions. In addition, it manages the availability of discourse entities, moving entities in and out of focus as the discourse proceeds from one segment to the next.

IMPLEMENTATION

The TFR Discourse Manager is implemented as a single top-level control module, written in Prolog, which uses PUNDIT as a resource. Its highest-level goals are to collect pre-defined information from the user and send the resulting information state to a database update module.

At the level of user interaction, the module's goals are to process the request-response units corresponding to the header items and the segments A-E. In the header segment, the Discourse Manager prompts for each of the header items (speaker, date, part number, etc.), and calls PUNDIT to analyze the responses. The responses give rise to discourse entities, whose representations are added to the DISCOURSE LIST for subsequent full-NP reference. The three global foci (speaker, failed part, and equipment) are stored in a distinguished location in the discourse model.

For each of the remaining segments (A-E), the processing is described below.

1. Initialize Discourse Context

At the start of each segment, we empty the list of salient entities from the previous segment (the FOCUS LIST) and load in the global foci. This prevents pronominal reference from crossing segment boundaries (although full NP reference is possible).

2. Prompt the User

Before the system can interpret the user's response to a prompt, it must first 'understand' what it is about to ask. This step, while intuitive, is actually required in order to create the context for interpreting the response. We look up the meaning of the prompt (stored as a lambda expression), create a discourse entity, and place it at the head of the focus list. This makes the prompt the most salient entity in the context when the response is processed, and allows for both pronominal and implicit reference, e.g. **Probable cause: UNKNOWN**. Having done this, we issue the prompt and collect the user's response.

3. Analyze the Response

Two levels of interpretation are provided. First, PUNDIT is called to analyze the response; next, the response entity is bound to a variable in the representation of the prompt, to derive a new proposition.

Two types of call to PUNDIT are required, in order to handle both NP responses (**BROKEN WIRE**) and sentential or paragraph responses (**BELIEVE PROBLEM TO HAVE BEEN CAUSED BY FAILURE OF UPPER WIDGET**). If the response can be analyzed by PUNDIT's syntactic component as an NP, then a side-door to PUNDIT semantic and pragmatic analysis is used to provide a semantic interpretation and create a discourse entity.

If the response cannot be analyzed as an NP, then the normal entrance points for syntactic and semantic/pragmatic analysis are used. This results in the creation of one or more situation entities, which are grouped together to form a higher-level response entity.

Finally, the response entity is bound to the variable in the representation of the prompt, and lambda reduction is applied. The resulting representation is added to the discourse list, where it becomes available for subsequent full-NP reference (e.g. *The failure...*, *The cause...*).

RESEARCH DIRECTIONS

The implementation described above partially captures our observations concerning the discourse structure of TFRs and how it constrains pronominal reference, as well as the discourse relation of requests to responses. It thus provides a level of discourse management and interpretation beyond that developed for previous PUNDIT applications. Our experience with this application has led us in two research directions: towards the management of open-ended dialogue, and towards the development of a domain-independent discourse interpretation facility.

References

- [Blair 85] Blair, David C. and Marion, M. E. An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System. *Communications of the ACM* 28(3):289–299, 1985.
- [Dahl 86] Dahl, Deborah A. Focusing and Reference Resolution in PUNDIT. In *Proceedings of the 5th National Conference on Artificial Intelligence*. Philadelphia, PA, August 1986.
- [Dahl 87] Dahl, Deborah A., Dowding, John, Hirschman, Lynette, Lang, Francois, Linebarger, Marcia, Palmer, Martha, Passonneau, Rebecca, and Riley, Leslie. *Integrating Syntax, Semantics, and Discourse: DARPA Natural Language Understanding Program, R and D Status Report*. Technical Report, Unisys Corporation, May 1987.
- [Delahunty 82] Delahunty, Gerald P. *Topics in the Syntax and Semantics of English Cleft Sentences*. Indiana University Linguistics Club, Bloomington, 1982.
- [Grishman 86a] Grishman, Ralph and Hirschman, Lynette. PROTEUS and PUNDIT: Research in Text Understanding. *Computational Linguistics* 12(2):141–45, 1986.
- [Grishman 86b] Grishman, Ralph and Kitteridge, Richard (editors). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum, New Jersey, 1986.
- [Grosz 86] Grosz, Barbara J. and Sidner, Candace L. Attention, Intentions and the Structure of Discourse. *Computational Linguistics*, 1986.
- [Higgins 79] Higgins, F. R. *The Pseudo-Cleft Construction in English*. Garland, 1979.
- [Hirschberg 87] Hirschberg, Julia and Litman, Diane. Now Let's Talk about *now*: Identifying Cue Phrases Intonationally. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 163–171. Stanford, CA, July 1987.
- [Linebarger 88] Linebarger, Marcia C., Dahl, Deborah A., Hirschman, Lynette, and Passonneau, Rebecca J. Sentence Fragments Regular Structures. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*. Buffalo, NY, June 1988.
- [Prince 86] Prince, Ellen F. On the Syntactic Marking of Presupposed Open Propositions. In *Papers from the Parasession on Pragmatics and Grammatical Theory at the 22nd Regional Meeting of the Chicago Linguistic Society*. 1986.
- [Schiffrin 87] Schiffrin, Deborah. *Discourse Markers*. Cambridge University Press, Cambridge, 1987.
- [Webber 88] Webber, Bonnie Lynn. Discourse deixis: Reference to Discourse Segments. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 113–122. Buffalo, NY, June 1988.